
Lens-LeJEPa: Learning the Right Invariances for Strong Gravitational Lenses

Arnesh Batra^{1*} Karthik Gaur² Pranath Reddy³ Michael W. Toomey⁴ Sergei Gleyzer²

¹Indraprastha Institute of Information Technology, Delhi ²The University of Alabama

³University of Alabama at Huntsville ⁴Massachusetts Institute of Technology

Abstract

Strong gravitational lenses probe dark matter on small scales, but labelled simulations are expensive and labelled real lenses are rare. Self-supervised pretraining should help, yet its standard augmentations are physically wrong here: zooming a lens image rescales the Einstein radius and with it the enclosed mass. **Lens-LeJEPa** replaces them with two lensing priors on a teacher-free joint-embedding objective: exact D_4 correspondence between image patches, and consistency of arc-weighted descriptors pooled in lens-centred rings. One encoder, pretrained on 87.5k unlabelled images and then frozen, serves three tasks through low-rank adapters. With the adapter held fixed, the priors add 11.0 accuracy points at 100 labels per class, and Lens-LeJEPa beats the previous lensing-specific JEPa by 13 to 30 points; rank-stabilized LoRA matches full fine-tuning while training 11% of the weights. The same invariance that helps classification limits axion-mass regression and reconstruction, and we trace both outcomes to one mechanism.

1 Introduction

A massive galaxy between us and a distant source bends its light into arcs whose shape depends on how the lens mass is distributed, making strong lenses one of the few probes of dark matter below galactic scales [Vegetti et al., 2010]. Dark matter models leave different marks: cold dark matter adds discrete subhaloes, while an ultra-light axion-like particle forms vortex-like defects [Hu et al., 2000, Alexander et al., 2020a]. Supervised networks separate these well on simulations, but they need large label sets, which a simulation can only produce after assuming a dark matter model, while Euclid [Laureijs et al., 2011] will deliver many lens images and no labels. This is the setting self-supervised pretraining is built for, yet generic recipes transfer poorly: they rely on augmentations that are physically wrong here. Zooming rescales the apparent Einstein radius θ_E , which fixes the enclosed mass through $M \propto \theta_E^2$, so a zoom-invariant encoder is trained to discard a mass measurement.

Our contributions:

- **Lens-LeJEPa**, a teacher-free objective with two lensing priors, exact D_4 patch-token correspondence and arc-weighted ring consistency, computed from the image alone with no lens model and no labels.
- **A controlled comparison** in which the priors are the only difference between two otherwise identical runs: they help at every label budget, and Lens-LeJEPa beats Lens-JEPa by 13 to 30 accuracy points.
- **An adapter study** over seven families and a rank sweep, and **one frozen encoder on three tasks**, benchmarked against supervised baselines for each, with a single mechanism that explains where it wins and where it loses.

*Corresponding author: arnesh23129@iiitd.ac.in

2 Physics and data

The public DeepLense simulations [Toomey, 2021], generated with `lenstronomy` [Birrer and Amara, 2018, Birrer et al., 2021], place a sheared isothermal elliptical lens in front of a Sérsic source (Appendix B). The three classes, **axion** (vortex-like defects), **CDM** (point-mass subhaloes) and **no substructure**, differ only in the substructure added to the main halo, which appears as fine perturbations of the arcs rather than a change in the overall ring. **Model I** (150^2 px, 87.5k images) is our *only* pretraining source and none of its labels are used; **Model II** (Euclid-like) and **Model III** (HST-like), both 64^2 px with 15k test images, are downstream only, so every result is cross-instrument. Axion images carry the simulated particle mass (10^{-24} to 10^{-22} eV), and $\log_{10}(m/\text{eV})$ is our regression target. Two properties drive the design. The lens is centred and the point spread function is isotropic, so a 90° rotation or a reflection is an equally valid observation of the same system, with class, θ_E and mass unchanged. And the signal sits in thin arcs near θ_E .

3 Method

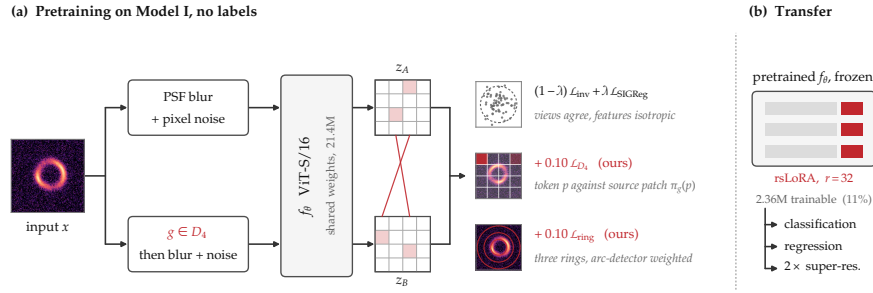


Figure 1: **Lens-LeJEPA. (1)** One ViT-S/16 encodes two views of an unlabelled Model I image that differ only by lens-safe perturbations and a D_4 transform; two of the three loss terms are ours. **(2)** The quantities the priors use, computed on a real image with the training code; none needs a label or a lens model. **(3)** The frozen encoder is adapted to each task with rsLoRA.

Base objective. We build on LeJEPA [Balestriero and LeCun, 2025], which learns from two views without a teacher, moving average or stop-gradient. Both views pass through the *same* ViT-S/16 [Dosovitskiy et al., 2021], giving patch-token maps $z_A, z_B \in \mathbb{R}^{P \times d}$ over the $P = 10 \times 10$ grid; mean-pooling and a small projector reduce them to vectors u_A, u_B . One term pulls the projections together, and a sketched isotropy term (SIGReg) keeps their distribution close to an isotropic Gaussian, which prevents collapse without a teacher, in the spirit of variance-covariance regularisation [Bardes et al., 2022]:

$$\mathcal{L}_{\text{base}} = (1 - \lambda) \underbrace{\frac{1}{V} \sum_v \|u_v - \bar{u}\|^2}_{\mathcal{L}_{\text{inv}}} + \lambda \mathcal{L}_{\text{SIGReg}}, \quad \lambda = 0.02, \quad (1)$$

with $\bar{u}$ the view mean and gradients through both branches. $\mathcal{L}_{\text{base}}$ sees only the pooled vectors; our priors act on the token maps, where the image-plane geometry survives.

Lens-safe views. A view must be an image the telescope could plausibly have returned for the *same* system. This admits a Gaussian blur of at most one native pixel (point-spread-function variation) and low-amplitude detector noise. It excludes what contrastive learning usually relies on: crop, zoom and rescale all change the apparent θ_E of a fixed system, so an encoder trained to ignore them ignores a mass measurement [Chang et al., 2023]. Translation is also excluded, because it moves the lens off centre and both priors are defined relative to that centre.

Prior 1: exact D_4 correspondence. The one geometric transformation that *is* safe is a change of detector orientation. We draw g from the dihedral group D_4 and apply it before forming the second view. Because g acts on a centred grid whose side is a multiple of the patch size, it maps whole patches onto whole patches and induces an exact permutation π_g of patch indices (proved and verified numerically in Appendix E). We can therefore compare tokens patch by patch instead of imposing a single global invariance:

$$\mathcal{L}_{D_4} = \text{smooth}_{L_1}(\text{LN}(z_B), \text{LN}(z_A[\pi_g])), \quad (2)$$

where LN is layer normalisation and $z_A[\pi_g]$ reindexes the canonical map. The term vanishes exactly when the encoder is D_4 -equivariant, so the network learns the equivariance that group-equivariant architectures hard-code [Cohen and Welling, 2016, Bronstein et al., 2021], with no change to the architecture.

Prior 2: arc-weighted ring consistency. Patch correspondence constrains *where* information sits, not *what* the arcs look like, and the substructure signal is a faint perturbation of arc morphology. We therefore pool each view’s tokens into $K = 3$ lens-centred rings, weighting each patch by a label-free arc detector $w(x)$, the image gradient plus Laplacian magnitude normalised to unit mean, so that patches on bright curved arcs dominate. With R_k the patches in ring k ,

$$\phi_k(z, x) = \frac{\sum_{p \in R_k} w_p(x) z_p}{\sum_{p \in R_k} w_p(x)}, \quad \mathcal{L}_{\text{ring}} = \text{smooth}_{L_1}(\text{LN}(\phi(z_B, x_B)), \text{LN}(\phi(z_A, x_A))). \quad (3)$$

Every element of D_4 preserves distance from the centre, so π_g maps each ring to itself (Appendix E) and ϕ is invariant by construction: matching it constrains arc morphology at each radius rather than restating the orientation constraint. The two priors are complementary, and the full objective is $\mathcal{L} = \mathcal{L}_{\text{base}} + 0.10 \mathcal{L}_{D_4} + 0.10 \mathcal{L}_{\text{ring}}$.

Transfer and protocol. The encoder is then frozen and every linear projection receives a rank-stabilized LoRA adapter [Kalajdziewski, 2023], which scales the update by $\alpha/\sqrt{r}$ instead of α/r so that its magnitude does not shrink as the rank grows. At $r=32$ this is 2.36M parameters, 11% of the encoder, and one backbone serves all three tasks. Lens-LeJEPA and LeJEPA are both 12-block ViT-S/16 at 160^2 , pretrained for 100 epochs on Model I with the same seed and checkpoint policy. Downstream runs use 50 epochs and the best-validation checkpoint, and every number is on held-out test data (Appendix C). I-JEPA and the Lens-JEPA family are references pretrained for 300 epochs ($3\times$ budget).

4 Results

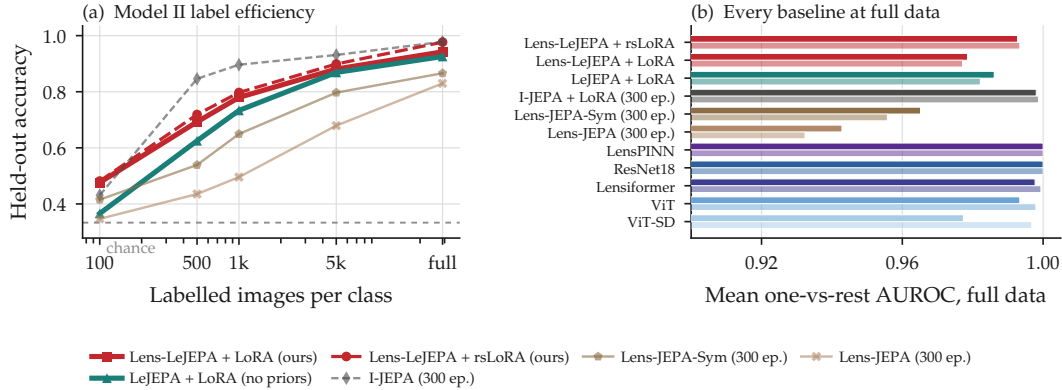
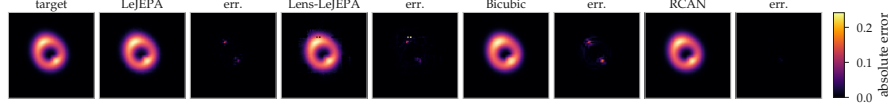


Figure 2: **Substructure classification.** (a) Label efficiency on Model II. Every curve uses a frozen encoder with a rank-32 adapter; the two LoRA curves of Lens-LeJEPA and LeJEPA differ only in the pretraining objective. (b) Every baseline at full data, mean one-vs-rest AUROC; darker bars are Model II, lighter bars Model III. The complete matrix at every budget is Appendix Table 6, and per-class values are Appendix Table 7.

4.1 Substructure classification

With the same LoRA adapter on both encoders (Figure 2a), the priors add +11.0 accuracy points at 100 labels per class, +6.8 at 500, +4.6 at 1k and +1.7 at full data, and +2.4 on Model III; the AUROC gain is likewise largest where labels are scarcest (+12.1 at 100). Switching to rsLoRA adds a further 0.5 to 3.5 points. Lens-LeJEPA beats Lens-JEPA by 13 to 30 points despite that model’s $3\times$ budget, and a soft *global* D_4 loss (Lens-JEPA-Sym) recovers only part of the gap. At full data (Figure 2b), Lens-LeJEPA with rsLoRA reaches AUROC 0.9927 on Model II and 0.9934 on Model III, ahead of the whole Lens-JEPA family and of ViT-SD on Model II; the other supervised models lead slightly on AUROC. The case for a pretrained encoder is label efficiency and reuse across tasks, not full-data accuracy.



Axion mass (MAE, dex $\downarrow$)			2 $\times$ SR (PSNR, dB $\uparrow$)		
Method	M-II	M-III	Method	M-II	M-III
Lens-LeJEPa + rsLoRA (ours)	<u>0.100</u>	<u>0.090</u>	<i>Frozen encoder + adapter</i>		
Lens-LeJEPa + LoRA (ours)	<u>0.149</u>	<u>0.109</u>	LeJEPa + LoRA	48.32	45.38
LeJEPa + LoRA	0.121	0.122	Lens-LeJEPa + LoRA (ours)	<u>44.70</u>	<u>42.50</u>
LensPINN (supervised)	0.042	0.035	<i>Supervised, from scratch</i>		
ResNet18 (supervised)	0.021	0.019	EDSR	67.07	62.62
			RCAN	65.12	61.41
			SRResNet	61.53	60.63
			FSRCNN	51.60	46.56
			Bicubic	41.07	40.27

Figure 3: **Downstream tasks.** Top: 2 $\times$ super-resolution on a held-out Model II lens, each reconstruction followed by its absolute error on a shared scale. Inputs are bicubic 64 $\rightarrow$ 32 downsamples, so SR measures reconstruction, not physical resolution recovery. All supervised networks are trained from scratch with the same 50-epoch budget. Best bold, ours underlined.

4.2 Axion mass regression

Estimating a continuous physical quantity is harder. The frozen encoder reaches 0.100 dex MAE and $R^2 = 0.937$ on Model II, the best self-supervised result on both datasets (Figure 3, lower left), but a supervised ResNet18 reaches 0.021 dex, about 4.8 $\times$ better. We attribute this to the objective rather than to tuning. A heavier axion has a shorter de Broglie wavelength, so its vortex pattern is finer and lower in contrast, and our objective explicitly asks for *invariance* to blur and noise in that same spatial band. The error profile agrees: it roughly quadruples from the lowest to the highest mass decile (Appendix Figure 9), that is, towards finer patterns.

4.3 Synthetic 2 $\times$ super-resolution

A light token-to-patch decoder on the frozen encoders reaches 48.32 dB (LeJEPa) and 44.70 dB (Lens-LeJEPa) on Model II, 7.2 and 3.6 dB above bicubic interpolation (Figure 3, lower right). Dedicated super-resolution networks trained from scratch on the same data and budget are far stronger: EDSR [Lim et al., 2017] reaches 67.07 dB, RCAN [Zhang et al., 2018] 65.12 dB and SRResNet [Ledig et al., 2017] 61.53 dB, and even the 12k-parameter FSRCNN [Dong et al., 2016] reaches 51.60 dB. Fine-tuning every encoder weight narrows the gap to 51.01 dB for Lens-LeJEPa, ahead of FSRCNN on Model III (Appendix Table 9). Pixel-level reconstruction is therefore where a frozen, invariance-trained encoder is weakest, and with a frozen encoder the priors *cost* 3.6 dB: the mirror image of the classification result.

4.4 Ablations

Table 1: **Adaptation capacity.** Held-out accuracy and per-step cost (batch 64, RTX A4500). Best values bold, ours underlined. Lens-LeJEPa has no frozen-probe run, so the probe comparison uses the reference encoder.

	Train.	ms /	Model II accuracy by label budget					Model III
Adaptation	(M)	step	100	500	1k	5k	full	full
<i>Lens-LeJEPa encoder (ours)</i>								
LoRA $r=32$	2.36	121	0.4763	0.6927	0.7797	0.8809	0.9423	0.9453
rsLoRA $r=32$ (ours)	2.36	122	0.4815	0.7188	0.7972	0.8989	0.9768	0.9744
Full fine-tuning	21.39	102	0.4851	0.7150	0.8084	0.8979	0.9751	0.9776
<i>Reference I-JEPa encoder: what an adapter buys over a probe</i>								
Frozen linear probe	0.001	35	0.3848	0.4121	0.4351	0.4884	0.6002	0.6235
LoRA $r=32$	2.36	121	0.4311	0.8461	0.8963	0.9311	0.9780	0.9905

The two priors. Removing both priors gives exactly the LeJEPa curve in Figure 2: with the same adapter, this costs 1.2 to 11.0 accuracy points on Model II and 2.4 on Model III, while the pooled effective-rank fraction rises from 0.164 to 0.213 (Appendix Figure 5). Replacing exact patch correspondence with a global symmetry loss recovers only part of the gap, because a pooled loss discards the exact correspondence π_g provides (Appendix E).

Adaptation capacity. On the reference encoder a frozen linear probe reaches only 0.6002 at full data, against 0.9780 with LoRA (Table 1). Full fine-tuning gains at most 1.2 points over rsLoRA

and loses on three of six settings, while training $9\times$ more parameters. rsLoRA saves trainable parameters and optimiser state, *not* step time (122 against 102 ms/step at batch 64). Across seven adapter families (Appendix Table 4), rsLoRA is best in all six settings.

5 Discussion and conclusion

One mechanism explains all three results. The lensing priors make the encoder invariant to detector orientation and to blur and noise, which suits semantic tasks and hurts detail-sensitive ones. Sub-structure class does not depend on orientation, so classification gains up to 11 points over the same objective without priors, with the largest gains when labels are scarce. Reconstruction needs exactly the pixel-level detail these invariances discard, so the frozen encoder loses 3.6 dB, and mass regression stays $4.8\times$ behind supervised training because blur invariance suppresses the finest texture. The lower pooled effective rank, 0.164 against 0.213, is the same compression seen in the representation.

The broader lesson is that the choice of invariances is itself a physical modelling decision. Generic self-supervised recipes encode assumptions from natural images, such as invariance to scale and framing, that are false for lenses; replacing them with symmetries the instrument actually guarantees, and imposing those symmetries through the loss rather than the architecture, keeps a standard ViT and its tooling while adding only two cheap, label-free terms. Because every invariance also discards information, it should be matched to the downstream quantity: the geometry-aware encoder is the right choice for classification and lens discovery, where labels are scarce and the target is orientation-free, while the encoder without priors, or full fine-tuning, suits continuous quantities and pixel-level tasks. A natural extension is to make the invariance task-dependent, for example by keeping a branch without blur invariance for mass estimation. The main open question is transfer to real survey data, where lenses are imperfectly centred and lens-light subtraction leaves residuals; Appendix H discusses this and other limitations.

References

- Stephon Alexander, Sergei Gleyzer, Evan McDonough, Michael W. Toomey, and Emanuele Usai. Deep learning the morphology of dark matter substructure. *The Astrophysical Journal*, 893(1):15, 2020a. arXiv:1909.07346.
- Stephon Alexander, Sergei Gleyzer, Hanna Parul, Pranath Reddy, Michael W. Toomey, Emanuele Usai, and Ryker Von Klar. Decoding dark matter substructure without supervision. *arXiv preprint arXiv:2008.12731*, 2020b.
- Mahmoud Assran, Quentin Duval, Ishan Misra, Piotr Bojanowski, Pascal Vincent, Michael Rabbat, Yann LeCun, and Nicolas Ballas. Self-supervised learning from images with a joint-embedding predictive architecture. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15619–15629, 2023. arXiv:2301.08243.
- Randall Balestriero and Yann LeCun. LeJEPa: Provable and scalable self-supervised learning without the heuristics. *arXiv preprint arXiv:2511.08544*, 2025.
- Adrien Bardes, Jean Ponce, and Yann LeCun. VICReg: Variance-invariance-covariance regularization for self-supervised learning. In *International Conference on Learning Representations (ICLR)*, 2022. arXiv:2105.04906.
- Simon Birrer and Adam Amara. lenstronomy: Multi-purpose gravitational lens modelling software package. *Physics of the Dark Universe*, 22:189–201, 2018. arXiv:1803.09746.
- Simon Birrer, Anowar J. Shajib, Daniel Gilman, Aymeric Galan, Jelle Aalbers, Martin Millon, Robert Morgan, Giulia Pagano, Ji Won Park, Luca Teodori, Nicolas Tessore, Madison Ueland, Lyne Van de Vyvere, Sebastian Wagner-Carena, Ewoud Wempe, Lilan Yang, Xuheng Ding, Thomas Schmidt, Dominique Sluse, Ming Zhang, and Adam Amara. lenstronomy II: A gravitational lensing software ecosystem. *Journal of Open Source Software*, 6(62):3283, 2021.
- Michael M. Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *arXiv preprint arXiv:2104.13478*, 2021.
- Po-Wen Chang, Kuan-Wei Huang, Joshua Fagin, James Hung-Hsu Chan, and Joshua Yao-Yu Lin. LenSiam: Self-supervised learning on strong gravitational lens images. In *NeurIPS 2023 AI for Science Workshop*, 2023. arXiv:2311.10100.

- Taco S. Cohen and Max Welling. Group equivariant convolutional networks. In *Proceedings of the 33rd International Conference on Machine Learning (ICML)*, 2016. arXiv:1602.07576.
- Chao Dong, Chen Change Loy, and Xiaoou Tang. Accelerating the super-resolution convolutional neural network. In *European Conference on Computer Vision (ECCV)*, 2016. arXiv:1608.00367.
- Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations (ICLR)*, 2021. arXiv:2010.11929.
- Soufiane Hayou, Nikhil Ghosh, and Bin Yu. LoRA+: Efficient low rank adaptation of large models. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235 of *PMLR*, pages 17783–17806, 2024. arXiv:2402.12354.
- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.
- Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022. arXiv:2106.09685.
- Wayne Hu, Rennan Barkana, and Andrei Gruzinov. Cold and fuzzy dark matter. *Physical Review Letters*, 85:1158–1161, 2000. arXiv:astro-ph/0003365.
- Nam Hyeon-Woo, Moon Ye-Bin, and Tae-Hyun Oh. FedPara: Low-rank hadamard product for communication-efficient federated learning. In *International Conference on Learning Representations (ICLR)*, 2022. arXiv:2108.06098.
- Damjan Kalajdzievski. A rank stabilization scaling factor for fine-tuning with LoRA. *arXiv preprint arXiv:2312.03732*, 2023.
- Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. Fast and accurate image super-resolution with deep laplacian pyramid networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(11):2599–2613, 2019. arXiv:1710.01992.
- R. Laureijs, J. Amiaux, S. Arduini, J.-L. Auguères, J. Brinchmann, R. Cole, M. Cropper, C. Dabin, L. Duvet, A. Ealet, et al. Euclid definition study report. *arXiv preprint arXiv:1110.3193*, 2011.
- Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. arXiv:1609.04802.
- Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2017. arXiv:1707.02921.
- Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, Pavlo Molchanov, Yu-Chiang Frank Wang, Kwang-Ting Cheng, and Min-Hung Chen. DoRA: Weight-decomposed low-rank adaptation. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2024. arXiv:2402.09353.
- Ashutosh Ojha, Sergei Gleyzer, Michael W. Toomey, and Pranath Reddy Kumbam. LensPINN: Physics informed neural network for learning dark matter morphology in lensing. In *NeurIPS 2024 Workshop on Machine Learning and the Physical Sciences*, 2024. https://ml4physicalsciences.github.io/2024/files/NeurIPS_ML4PS_2024_78.pdf.
- Achmad Ardani Prasha, Clavino Ourizqi Rachmadi, Muhamad Fauzan Ibnu Syahlan, Naufal Rahfi Anugerah, Nanda Garin Raditya, Putri Amelia, Sabrina Laila Mutiara, and Hilman Syachr Ramadhan. Masked autoencoder pretraining on strong-lensing images for joint dark-matter model classification and super-resolution. *arXiv preprint arXiv:2512.06642*, 2025.

- J. Rishi, Pranath Reddy Kumbam, Michael W. Toomey, and Sergei Gleyzer. Lens-JEPA: Physics informed joint embedding predictive architecture for gravitational lensing. In *NeurIPS 2025 Workshop on Machine Learning and the Physical Sciences*, 2025. https://ml4physicalsciences.github.io/2025/files/NeurIPS_ML4PS_2025_340.pdf.
- George Stein, Jacqueline Blaum, Peter Harrington, Tomislav Medan, and Zarija Lukić. Mining for strong gravitational lenses with self-supervised learning. *The Astrophysical Journal*, 932(2):107, 2022. arXiv:2110.00023.
- Michael W. Toomey. DeepLenseSim: Software to simulate strong lensing images for the DeepLense project. <https://github.com/mwt5345/DeepLenseSim>, 2021.
- Simona Vegetti, L. V. E. Koopmans, Adam S. Bolton, Tommaso Treu, and Raphaël Gavazzi. Detection of a dark substructure through gravitational imaging. *Monthly Notices of the Royal Astronomical Society*, 408(4):1969–1981, 2010. arXiv:0910.0760.
- Lucas J. Veloso, Michael W. Toomey, and Sergei Gleyzer. Lensformer: A physics-informed vision transformer for gravitational lensing. In *NeurIPS 2023 Workshop on Machine Learning and the Physical Sciences*, 2023. https://ml4physicalsciences.github.io/2023/files/NeurIPS_ML4PS_2023_214.pdf.
- Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.
- Shih-Ying Yeh, Yu-Guan Hsieh, Zhidong Gao, Bernard B. W. Yang, Giyeong Oh, and Yanmin Gong. Navigating text-to-image customization: From LyCORIS fine-tuning to model evaluation. In *International Conference on Learning Representations (ICLR)*, 2024. arXiv:2309.14859.
- Qingru Zhang, Minshuo Chen, Alexander Bukharin, Nikos Karampatziakis, Pengcheng He, Yu Cheng, Weizhu Chen, and Tuo Zhao. AdaLoRA: Adaptive budget allocation for parameter-efficient fine-tuning. In *International Conference on Learning Representations (ICLR)*, 2023. arXiv:2303.10512.
- Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. arXiv:1807.02758.

A Related work

Substructure from lens images. Machine learning here began with supervised classifiers [Alexander et al., 2020a] and unsupervised clustering [Alexander et al., 2020b]. The strongest current baselines fold the lens equation into the network: Lensformer [Veloso et al., 2023] adds a branch approximating the lensing potential and LensPINN [Ojha et al., 2024] extends it with physics-informed preprocessing. Both are supervised, and both lead us at full data.

Self-supervised lensing models. Contrastive pretraining has been used to mine surveys for lens candidates [Stein et al., 2022], and LenSiam [Chang et al., 2023] made the observation this paper builds on: crop and zoom change the Einstein radius, destroying the quantity a lensing model should measure. **Lens-JEPA** [Rishi et al., 2025] is the closest work and our main comparison. It extends I-JEPA [Assran et al., 2023], which predicts masked target blocks in latent space with a moving-average teacher, by adding a physics encoder guided by the lens equation. A concurrent masked-autoencoder study pairs classification with super-resolution here [Prasha et al., 2025].

What we do differently. All of the above put the physics, or the prediction machinery, into the *architecture*. We put it into the *objective*, on the teacher-free LeJEPA formulation [Balestrierio and LeCun, 2025], so each prior is one additive term whose removal gives an exact ablation, and the encoder stays an ordinary ViT with no teacher, moving average or mask sampler.

B Data and physics details

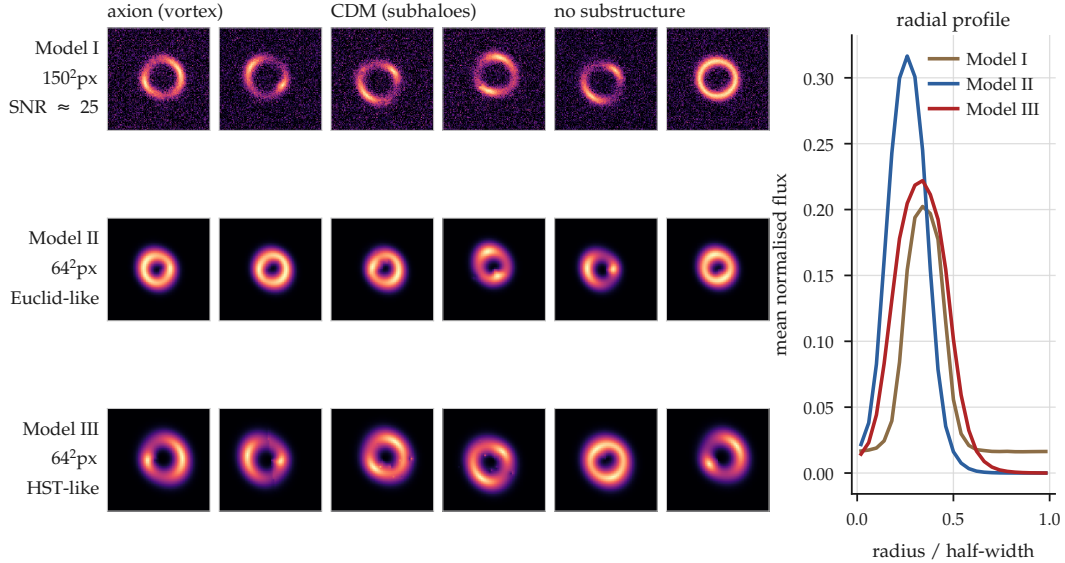


Figure 4: **The three DeepLense datasets.** Two held-out examples per class per dataset, shown with an arcsinh stretch. The class difference is a perturbation of the arc, not a change of the overall ring: axion images carry vortex-like interference structure, CDM images carry discrete subhalo perturbations, and the no-substructure images have a smooth arc. The right panel gives the azimuthally averaged radial profile of each dataset. Model I is noisier and broader, Model II (Euclid-like) is the most concentrated, and Model III (HST-like) sits between them; this is the domain shift the frozen encoder must survive, since pretraining only ever sees Model I.

Simulation setup and attribution. All images come from the public DeepLense simulations, produced by the DeepLenseSim pipeline [Toomey, 2021] on top of lenstronomy [Birrer and Amara, 2018, Birrer et al., 2021]; the substructure models they implement are those of Alexander et al. [2020a,b]. We cite the pipeline and the software separately because the datasets have no paper of their own, and we follow lenstronomy’s own attribution policy in citing both software references. The lens is a sheared isothermal elliptical mass distribution and the background source follows a Sérsic profile. Substructure is added to the base halo to create the three classes. Model I applies a Gaussian point spread function with Gaussian and Poisson noise at signal-to-noise ≈ 25 at 150×150 pixels; Model II applies Euclid-like observing characteristics at 64×64 ; Model III applies HST-like characteristics at 64×64 .

Class counts. Model I: 28,897 axion / 29,772 CDM / 28,856 no-substructure (87,525 unlabelled pretraining images). Model II: 29,896 / 29,759 / 29,449 training and 5,000 per class held-out test. Model III: 29,812 / 29,584 / 30,000 training and 5,000 per class held-out test. Downstream validation is a stratified split of the training partition; the official test partitions are used once, for the final evaluation.

Why the priors are physically reasonable. The orientation of a lens on the sky is a matter of convention, the simulated point spread function is isotropic, and the lens sits at the centre of the frame. Applying an element of D_4 to the image is therefore equivalent to rotating or reflecting the detector, and produces an equally valid observation of the same system: the class, the Einstein radius θ_E and the axion mass are all unchanged. That is what makes exact patch correspondence a safe constraint rather than an assumption.

An image-plane crop, zoom or rescale is different in kind. It changes the *apparent* θ_E in pixels while the physical system is fixed, and θ_E determines the mass enclosed within it,

$$M(< \theta_E) = \frac{c^2}{4G} \frac{D_d D_s}{D_{ds}} \theta_E^2, \quad (4)$$

with D_d , D_s and D_{ds} the observer-lens, observer-source and lens-source angular diameter distances. Since $M \propto \theta_E^2$, a zoom-invariant encoder would be trained to discard a direct mass measurement, which is the opposite of what we want. Translation is excluded for a separate reason: it moves the lens off centre, and both the D_4 patch correspondence and the ring partition assume a centred lens.

Finally, the substructure signal is concentrated in thin arcs near θ_E , which is why a radius-organised, gradient-weighted descriptor is a useful summary, and why we weight patches by an image-gradient and Laplacian response rather than by a learned or label-dependent saliency.

Regression target. Only axion images carry a simulated mass, and it is the mass of the ultra-light *particle*, not of a halo: the stored values span 10^{-24} to 10^{-22} eV, the fuzzy dark matter scale [Hu et al., 2000]. The target is therefore $\log_{10}(m/\text{eV})$, the normaliser is fit on the training split alone, and splits are deterministic mass quantiles so the full target range appears in train, validation and test.

Super-resolution task definition. The native 64×64 image is the target. The input is that image after antialiased bicubic downsampling to 32×32 , resized back to 160×160 for the encoder. The task is therefore explicitly synthetic image reconstruction. The DeepLense release does not contain a paired low-/high-resolution physical simulation for Models II and III, so no physical de-lensing or resolution-recovery claim is made anywhere in this paper.

C Implementation and training details

Related DeepLense work. Beyond the supervised classifiers used as baselines here, unsupervised clustering of substructure has also been explored on this simulation family [Alexander et al., 2020b].

Concurrent work. A masked-autoencoder study on this benchmark also pairs dark matter classification with super-resolution [Prasha et al., 2025]. We do not tabulate it against our numbers: it uses a $16 \rightarrow 64$ super-resolution factor rather than our $32 \rightarrow 64$, and a different split, so the metrics are not comparable.

Encoders. Both self-supervised encoders are standard 12-block ViT-S/16 (384 dimensions, 6 heads, 21.43M parameters) at 160×160 input with 16×16 patches, giving a 10×10 token grid. Native ViT initialisation including residual-layer rescaling is preserved. The LeJEPa projector is a two-layer BatchNorm MLP to 256 dimensions; both views are projected in one batch so they share normalisation statistics.

Pretraining. 100 epochs on Model I only, effective batch 128, AdamW, peak learning rate 5×10^{-4} with 10 warm-up epochs and cosine decay, weight decay 0.05, bfloat16, 256 SIGReg slices, $\lambda = 0.02$, lens prior weights 0.10 each, three radial rings, arc-detector strength 0.75, blur probability 0.5 with $\sigma \leq 1$ native pixel, and noise standard deviation up to 0.03. Every epoch records

feature standard deviation, effective-rank fraction, inter-view cosine, gradient norm and finiteness; a run that shows three consecutive epochs below the collapse thresholds is rejected rather than used downstream. Neither reported run triggered this.

Downstream. Frozen encoder, adapters on all 48 linear projections, 50 epochs, batch 128, AdamW at 3×10^{-4} with weight decay 0.01, validation every 10 epochs, seed 42, and evaluation of the best-validation checkpoint only. Classification adds label smoothing 0.1; regression uses the normalised scalar target; super-resolution uses Charbonnier plus SSIM loss [Lai et al., 2019, Wang et al., 2004] with a token-to-patch decoder.

Adapter families. *LoRA* [Hu et al., 2022] scales by α/r ; *rsLoRA* [Kalajdziewski, 2023] by $\alpha/\sqrt{r}$; *DoRA* [Liu et al., 2024] decomposes the adapted weight into a learned row magnitude and a normalised direction; *LoRA+* [Hayou et al., 2024] gives the B matrices twice the learning rate of A ; *gated LoRA* follows the adaptive-budget idea of AdaLoRA [Zhang et al., 2023] with a learned sigmoid gate per rank component; *LoHA* uses the low-rank Hadamard product of Hyeon-Woo et al. [2022] as popularised by Yeh et al. [2024]; *layerwise LoRA* varies rank with block depth.

Baselines. The supervised models are a ResNet18 [He et al., 2016], a ViT [Dosovitskiy et al., 2021], ViT-SD, Lensiformer [Veloso et al., 2023] and LensPINN [Ojha et al., 2024], each trained end to end on the labelled downstream set. The two reference self-supervised encoders, I-JEPA [Assran et al., 2023] and the Lens-JEPA family [Rishi et al., 2025], were pretrained for 300 epochs under an earlier protocol: I-JEPA on a 21.5M three-block wide ViT, Lens-JEPA and Lens-JEPA-Sym on a 2.45M single-block lensing backbone. They are labelled as $3\times$ budget references throughout and are not matched comparisons, which is why the controlled claim in this paper is always against LeJEPA at our own budget. For super-resolution we train four supervised networks from scratch, EDSR [Lim et al., 2017], RCAN [Zhang et al., 2018], SRResNet [Ledig et al., 2017] (the SRGAN generator, trained without the adversarial loss) and the lightweight FSRCNN [Dong et al., 2016], on the same low-resolution images and targets, with the same loss and 50-epoch budget as the encoder decoders, and report bicubic interpolation as a training-free reference.

Compute. Pretraining each encoder takes roughly 6 GPU-hours on a single RTX A4500; the complete set of downstream runs reported here is about 60 GPU-hours.

D Pretraining diagnostics

Both runs are single-seed, so the case that they are comparable rests on their training curves rather than on repetition. Figure 5 shows the nine quantities recorded every epoch. Three points matter. Neither run collapses: feature standard deviation rises and then plateaus, and the effective-rank fraction stays well above the 0.04 rejection threshold throughout, so neither checkpoint was accepted under a rescue. The two lensing terms converge rather than fight the base objective, falling by roughly two orders of magnitude, 1.9×10^{-2} to 1.5×10^{-4} for $\mathcal{L}_{D_4}$ and 6.7×10^{-3} to 1.3×10^{-4} for $\mathcal{L}_{\text{ring}}$, while the invariance and SIGReg terms follow the same trajectory in both runs. And the cost of the priors is visible directly in the representation: Lens-LeJEPA ends with a lower pooled effective-rank fraction, 0.164 against 0.213, and a lower feature standard deviation, 0.236 against 0.779. That is the invariance trade-off of Section 5, measured on the features rather than inferred from the downstream metrics.

E Why the patch permutation is exact

The D_4 term is worth stating precisely, because its value rests on a claim that is either exactly true or not true at all: that a D_4 transform of the image induces an *exact* permutation of patch indices, so the loss compares tokens of genuinely corresponding patches rather than merely encouraging a global invariance.

Setup. Let the image be $x \in \mathbb{R}^{N \times N}$ with $N = PG$, so that patch extraction $E : \mathbb{R}^{N \times N} \rightarrow \mathbb{R}^{G^2 \times P \times P}$ cuts x into a $G \times G$ grid of $P \times P$ blocks, $E(x)_p$ being block $p = (i, j)$ in row-major order. Write $D_4 = \{r^k, r^k s : k = 0, \dots, 3\}$ for the dihedral group of the square, acting on the

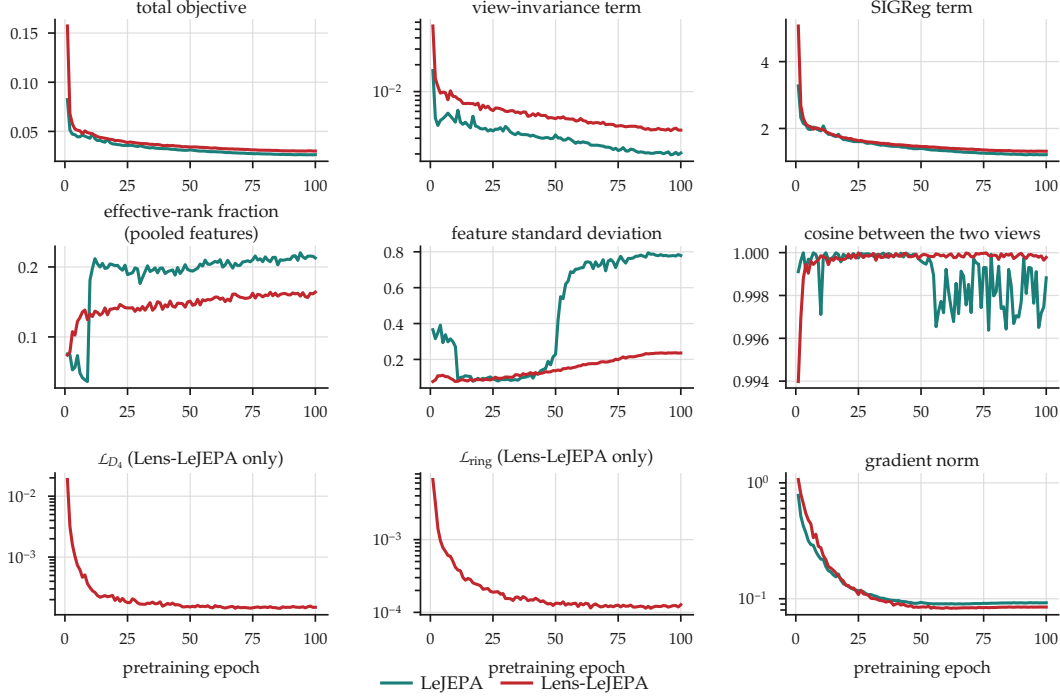


Figure 5: **Pretraining diagnostics over 100 Model I epochs.** Both runs are stable and neither collapses. The two lensing terms fall by roughly two orders of magnitude, so the constraints are satisfied rather than fought. The cost is visible in the middle row: Lens-LeJEPA ends with a lower pooled effective-rank fraction (0.164 vs 0.213) and lower feature standard deviation (0.236 vs 0.779) than the same objective without the priors. This is the quantitative version of the invariance trade-off discussed in the main text.

image about its centre: r is a quarter turn and s a reflection. In our setting $N = 160$, $P = 16$, $G = 10$.

Proposition 1 (Exact block permutation). *For every $g \in D_4$ there is a permutation π_g of $\{0, \dots, G^2 - 1\}$ such that, for all x and all p ,*

$$E(g \cdot x)_p = g \cdot E(x)_{\pi_g(p)}, \quad (5)$$

where the g on the right acts inside a single $P \times P$ block. Moreover $\pi : g \mapsto \pi_g$ is an injective group homomorphism $D_4 \rightarrow S_{G^2}$.

Proof. Index a pixel by $u = (u_1, u_2) \in \{0, \dots, N - 1\}^2$ and decompose it uniquely as $u = Pq + r$ with block index $q \in \{0, \dots, G - 1\}^2$ and offset $r \in \{0, \dots, P - 1\}^2$. It suffices to check the two generators. For the quarter turn, $(r \cdot x)(u_1, u_2) = x(u_2, N - 1 - u_1)$. Substituting $u = Pq + r$ and using $N - 1 = PG - 1 = P(G - 1) + (P - 1)$ gives

$$N - 1 - u_1 = P((G - 1) - q_1) + ((P - 1) - r_1),$$

in which the first term is a multiple of P and the second lies in $\{0, \dots, P - 1\}$. So this is again a valid block/offset decomposition, with block index $(q_2, (G - 1) - q_1)$ and offset $(r_2, (P - 1) - r_1)$. The block index depends only on q and the offset only on r : the action factors over the decomposition, which is exactly (5) with $\pi_r(q_1, q_2) = (q_2, (G - 1) - q_1)$. The same substitution for the reflection $(s \cdot x)(u_1, u_2) = x(u_1, N - 1 - u_2)$ gives $\pi_s(q_1, q_2) = (q_1, (G - 1) - q_2)$. Because both generators act on the block index alone, so does every word in them, and π is a homomorphism; it is injective because $\pi_g = \text{id}$ forces g to fix every block, hence to be the identity whenever $G \geq 2$. $\square$

Two consequences matter for the method. First, $\mathcal{L}_{D_4}$ is well posed: by (5), token p of the transformed view and token $\pi_g(p)$ of the canonical view see the same pixels up to the in-block action, so the loss is comparing physically corresponding patches, not an arbitrary spatial pairing. A global invariance

loss on pooled features, as in Lens-JEPA-Sym, discards this correspondence, which is what the ablation in Section 4.4 measures. Second, the loss vanishes exactly on the set of D_4 -equivariant encoders, that is those with $f_\theta(g \cdot x) = \pi_g \cdot f_\theta(x)$ up to the layer normalisation.

What is exact and what is learned. Proposition 1 is a statement about *patch geometry*, and it holds regardless of the network. It is not a claim that our encoder is equivariant. A plain ViT is not: its fixed sine-cosine positional embeddings are not D_4 -symmetric, so equivariance is the target that $\mathcal{L}_{D_4}$ optimises towards rather than a property the architecture enforces, unlike hard-wired group-equivariant networks [Cohen and Welling, 2016]. The measured residual is small but nonzero: the term falls from 1.9×10^{-2} at epoch 1 to 1.5×10^{-4} at epoch 100 (Figure 5).

Proposition 2 (The ring descriptor is D_4 -invariant by construction). *Let $R_k = \{q : \lfloor \rho(q)K \rfloor = k\}$ be the ring partition, where $\rho(q)$ is the distance from q to the grid centre normalised by the half diagonal. Then $\pi_g(R_k) = R_k$ for every $g \in D_4$ and every k .*

Proof. Both generators computed above are isometries of the grid that fix its centre $(\frac{G-1}{2}, \frac{G-1}{2})$: π_r is a rotation by a quarter turn about that point and π_s a reflection in a line through it. Hence $\rho(\pi_g(q)) = \rho(q)$ for all q , so π_g preserves each level set of $\lfloor \rho K \rfloor$. $\square$

Consequently ϕ_k pools over a set that the group action leaves fixed, so matching ϕ across views adds a constraint on *arc morphology at each radius* rather than re-imposing the orientation constraint that $\mathcal{L}_{D_4}$ already carries. The two priors are therefore not redundant.

Numerical validation. All four claims above are checked directly against the training code by `scripts/validate_theory.py`, which verifies, for $G = 10$ and $P = 16$: (5) pixel-for-pixel for all eight group elements; that each π_g is a bijection and π_e the identity; that $\pi_a \circ \pi_b = \pi_{ab}$ for all 64 pairs; and that the ring partition of Proposition 2 is invariant under all eight elements. All pass exactly.

Measured on one RTX A4500, the two lensing terms add 6% to an optimisation step, 202 against 191 ms. Our teacher-free step is dearer than I-JEPA’s, 202 against 80 ms, because it runs two full-image encoder passes with gradients where I-JEPA runs one masked context pass and a no-grad teacher. I-JEPA needs $3\times$ the epochs, however, so the totals are comparable: 7.7 hours against 9.1.

Table 2: **Measured pretraining cost**, one RTX A4500, batch 64, median over 12 interleaved rounds with the repository’s own modules. Epoch and total times extrapolate the measured step to the 87.5k-image Model I epoch at the epoch budget each method actually used. No wall-clock time was recorded during the original runs, so these were measured separately.

Objective	Epochs	ms / step	min / epoch	Total (h)
LeJEPA	100	191	4.4	7.3
Lens-LeJEPA (ours)	100	202	4.6	7.7
I-JEPA	300	80	1.8	9.1

F Additional classification results

This section reports everything the main text summarises, so that no comparison is hidden by the selection used in Figure 2.

Representation quality without an adapter. Table 3 isolates the encoders from the adaptation method: a frozen linear probe has no capacity of its own, so it measures the representation directly. Only the 300-epoch reference sweep was run this way, and neither matched-budget encoder has a probe run, so this table compares I-JEPA against the Lens-JEPA family rather than against our own encoder. Two things stand out. Every probe is far below what the same encoder reaches with an adapter, I-JEPA going from 0.6002 to 0.9780 at full data, which is why we report adapters throughout. And the ordering under a probe is not the ordering under an adapter: Lens-JEPA-Sym leads the probe table at four of six settings, yet Table 6 puts I-JEPA ahead of it by more than ten accuracy points once adapters are allowed. Probe rankings alone would have been misleading here.

Table 3: **Frozen linear probe**, held-out accuracy, no adapter. Best in each column is bold. Only the $3\times$ -budget reference encoders were run under this protocol.

Encoder	Pretrain	Model II accuracy by label budget					Model III
	epochs	100	500	1k	5k	full	full
I-JEPA	300	0.3848	0.4121	0.4351	0.4884	0.6002	0.6235
Lens-JEPA	300	0.3511	0.4101	0.4754	0.5460	0.6312	0.5883
Lens-JEPA-Sym	300	0.3710	0.4489	0.4687	0.5551	0.6443	0.6072

The complete few-shot matrix. Table 6 adds every model at every budget, including the ones Figure 2 omits. Read honestly it says: I-JEPA at $3\times$ the pretraining budget is ahead of Lens-LeJEPA between 500 and 5,000 labels, and LensPINN, which is supervised and physics-informed, is ahead of everything at every budget. Our claim is specifically about the matched-budget comparison and about the low-label end.

Table 4: **How to adapt a frozen lensing encoder.** Held-out accuracy under seven adapter families and full fine-tuning. rsLoRA is best on all six settings and matches full fine-tuning to within 1.2 accuracy points while training 11% of the weights.

Adaptation ($r=32$ unless noted)	Trainable	Model II accuracy by label budget					Model III	Mean
		100	500	1k	5k	full	full	
rsLoRA [Kalajdziewski, 2023]	2.36M	0.4815	0.7188	0.7972	0.8989	0.9768	0.9744	0.8079
LoRA+ [Hayou et al., 2024]	2.36M	0.4797	0.7077	0.7903	0.8877	0.9589	0.9646	0.7981
LoRA [Hu et al., 2022]	2.36M	0.4763	0.6927	0.7797	0.8809	0.9423	0.9453	0.7862
DoRA [Liu et al., 2024]	2.40M	0.4728	0.7044	0.7720	0.8842	0.9407	0.9409	0.7858
gated LoRA [Zhang et al., 2023]	2.36M	0.4751	0.7003	0.7775	0.8759	0.9426	0.9425	0.7856
layerwise-rank LoRA	2.07M	0.3757	0.6943	0.7756	0.8817	0.9441	0.9529	0.7707
LoHA [Hyeon-Woo et al., 2022]	4.72M	0.3592	0.5807	0.7561	0.8694	0.9192	0.9295	0.7357
full fine-tuning (no adapter)	21.4M	0.4851	0.7150	0.8084	0.8979	0.9751	0.9776	0.8098

Table 5: **LoRA rank**, chosen on the Model II *validation* split only and then fixed for every reported run. These are validation numbers and are not comparable to the held-out test tables.

LoRA rank r (with $\alpha = 2r$)	2	4	8	16	32	64
Trainable parameters	0.15M	0.29M	0.59M	1.18M	2.36M	4.72M
Model II <i>validation</i> accuracy	0.9248	0.9528	0.9663	0.9723	0.9814	0.9604

Table 6: **Complete Model II few-shot matrix**, including every reference encoder and every supervised baseline, so nothing that appears in the main text is selected for favourable comparison. I-JEPA at a $3\times$ pretraining budget leads our encoder between 500 and 5,000 labels, and LensPINN leads everywhere.

Method	Accuracy					Mean one-vs-rest AUROC				
	100	500	1k	5k	full	100	500	1k	5k	full
Lens-LeJEPA + rsLoRA (ours)	0.4815	0.7188	0.7972	0.8989	0.9768	0.6668	0.8475	0.9016	0.9604	0.9927
Lens-LeJEPA + LoRA (ours)	0.4763	0.6927	0.7797	0.8809	0.9423	0.6530	0.8457	0.8877	0.9664	0.9785
Lens-LeJEPA, full fine-tuning (ours)	0.4851	0.7150	0.8084	0.8979	0.9751	0.6721	0.8539	0.9140	0.9616	0.9932
LeJEPA + LoRA	0.3659	0.6251	0.7333	0.8688	0.9256	0.5317	0.7984	0.8702	0.9585	0.9861
I-JEPA + LoRA	0.4311	0.8461	0.8963	0.9311	0.9780	0.6275	0.9478	0.9752	0.9900	0.9981
Lens-JEPA-Sym + LoRA	0.4152	0.5390	0.6494	0.7969	0.8660	0.6034	0.7017	0.8144	0.9232	0.9652
Lens-JEPA + LoRA	0.3468	0.4353	0.4957	0.6792	0.8298	0.5301	0.6148	0.6747	0.8485	0.9428
LensPINN (supervised)	0.9097	0.9843	0.9945	0.9987	0.9997	0.9849	0.9995	0.9999	1.0000	1.0000
ResNet18 (supervised)	0.5999	0.8955	0.9863	0.9979	0.9999	0.7792	0.9787	0.9989	0.9999	1.0000
Lensformer (supervised)	0.6301	0.7540	0.9055	0.9401	0.9820	0.7878	0.8829	0.9706	0.9879	0.9977
ViT (supervised)	0.4989	0.7918	0.8321	0.8829	0.9397	0.6732	0.9207	0.9435	0.9687	0.9934
ViT-SD (supervised)	0.4243	0.7781	0.8305	0.8426	0.8960	0.5831	0.9081	0.9392	0.9483	0.9774

Which class is actually hard. Table 7 splits the AUROC by class. The no-substructure class is nearly saturated for every method, so the aggregate metric is dominated by the axion-versus-CDM distinction, which is the physically interesting one: both add small-scale structure to the arc and differ in its character rather than its presence. This is where the lensing priors help most.

Table 7: **Per-class one-vs-rest AUROC at full data.** The no-substructure class is easiest for every method. The hard pair is axion versus CDM, and this is where the lensing priors help most: rsLoRA raises the axion AUROC from 0.9694 to 0.9899 on Model II relative to standard LoRA on the same encoder.

Method	Model II one-vs-rest AUROC			Model III one-vs-rest AUROC		
	axion	CDM	no subs.	axion	CDM	no subs.
Lens-LeJEPa + rsLoRA (ours)	0.9899	0.9885	0.9998	0.9910	0.9896	0.9996
Lens-LeJEPa + LoRA (ours)	0.9694	0.9669	0.9992	0.9685	0.9636	0.9992
LeJEPa + LoRA	0.9825	0.9774	0.9984	0.9760	0.9722	0.9983
I-JEPa + LoRA	0.9972	0.9971	0.9999	0.9983	0.9980	0.9998
Lens-JEPa + LoRA	0.9391	0.8975	0.9920	0.9434	0.8768	0.9769
LensPINN (supervised)	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
ResNet18 (supervised)	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000

Adapters and rank in detail. The next three figures expand the ablation summarised in Section 4.4: the rank sweep and the adaptation regimes, then the full adapter-by-budget grid, then the held-out ROC and confusion structure of the selected configuration.

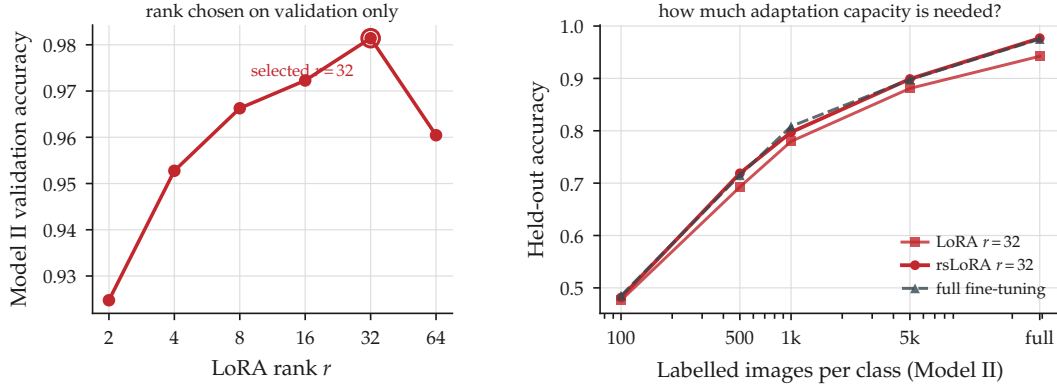


Figure 6: **Adaptation ablations.** Left: the LoRA rank search, run on the Model II validation split only, peaks at $r=32$ and degrades at $r=64$. Right: rsLoRA tracks full fine-tuning across the whole label range while training 11% of the weights, and both are far above a frozen linear probe on the same features.

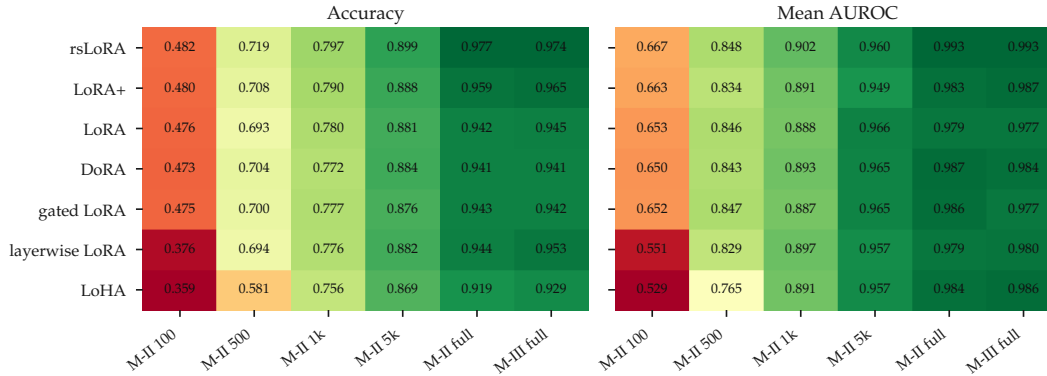


Figure 7: **Adapter family by label budget,** accuracy (left) and mean AUROC (right) for the frozen Lens-LeJEPa encoder. rsLoRA is the most consistent choice. The variants that reparameterise the update more aggressively (LoHA, layerwise) are the ones that break down at 100 labels.

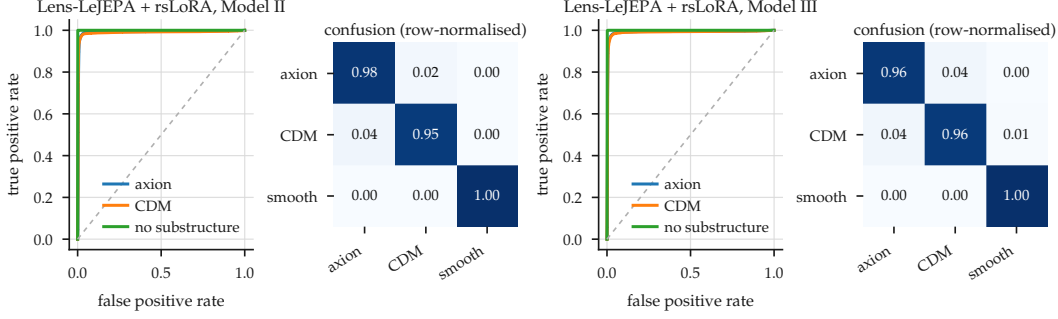


Figure 8: **Held-out ROC curves and confusion matrices** for Lens-LeJEPA + rsLoRA at full data. The residual confusion is almost entirely between axion and CDM; the smooth no-substructure class is separated cleanly on both instruments.

G Additional regression and super-resolution results

The main text reports one metric per task for space. Tables 8 and 9 give the rest, and the figures that follow show where the errors actually sit, which is more diagnostic than the aggregates.

Table 8: **Axion mass regression, all metrics.** The compact version in the main text reports MAE only.

Method	Model II			Model III		
	MAE (dex) ↓	RMSE (dex) ↓	R^2 ↑	MAE (dex) ↓	RMSE (dex) ↓	R^2 ↑
Lens-LeJEPA + rsLoRA (ours)	0.1001	0.1443	0.9374	0.0899	0.1265	0.9524
Lens-LeJEPA + LoRA (ours)	0.1487	0.2024	0.8768	0.1086	0.1505	0.9325
LeJEPA + LoRA	0.1213	0.1628	0.9203	0.1223	0.1614	0.9224
LensPINN (supervised)	0.0423	0.0633	0.9880	0.0351	0.0485	0.9930
ResNet18 (supervised)	0.0209	0.0374	0.9958	0.0192	0.0290	0.9975

Table 9: **Synthetic $2\times$ super-resolution, all metrics.** Every model is trained for 50 epochs (batch 128, AdamW at 3×10^{-4}) on the same data. The main text reports PSNR only.

Method	Model II			Model III		
	PSNR (dB) ↑	SSIM ↑	MAE ($\times 10^{-3}$) ↓	PSNR (dB) ↑	SSIM ↑	MAE ($\times 10^{-3}$) ↓
<i>Frozen self-supervised encoder + LoRA $r=32$ and token-to-patch decoder</i>						
LeJEPA + LoRA	48.32	0.99964	0.711	45.38	0.99902	0.985
Lens-LeJEPA + LoRA (ours)	44.70	0.99757	1.369	42.50	0.99815	1.572
<i>Self-supervised encoder, every weight fine-tuned</i>						
LeJEPA, full fine-tuning	50.88	0.99976	0.547	47.82	0.99915	0.848
Lens-LeJEPA, full fine-tuning (ours)	51.01	0.99967	0.558	48.23	0.99912	0.797
<i>Supervised super-resolution networks, trained from scratch at the same budget</i>						
EDSR [Lim et al., 2017]	67.07	0.99996	0.149	62.62	0.99989	0.205
RCAN [Zhang et al., 2018]	65.12	0.99995	0.166	61.41	0.99976	0.231
SRResNet [Ledig et al., 2017]	61.53	0.99993	0.232	60.63	0.99975	0.273
FSRCNN [Dong et al., 2016]	51.60	0.99886	0.819	46.56	0.99859	1.082
Bicubic upsampling (no training)	41.07	0.99867	1.921	40.27	0.99820	2.022

Where the regression error lives. Figure 9 shows that the frozen encoder tracks the target across the full mass range rather than failing in one region, but with visibly wider scatter than the supervised network, and that the residual grows towards the highest mass deciles. That is the regime where the vortex interference pattern is finest, and therefore the regime most affected by the blur invariance the objective enforces.

Super-resolution across both instruments. The last two figures give the metric bars for both test sets and a second qualitative panel on Model III, whose sharper arcs leave slightly larger residuals than Model II.

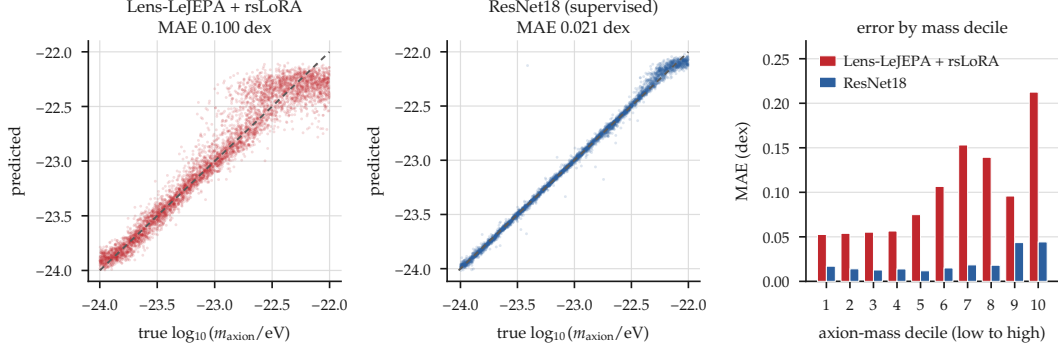


Figure 9: **Axion mass regression diagnostics on Model II.** The frozen encoder tracks the target over the full range but with visibly wider scatter than the supervised network, and the residual grows towards the highest mass deciles where the vortex pattern is finest, the regime most affected by the blur invariance the pretraining objective enforces.

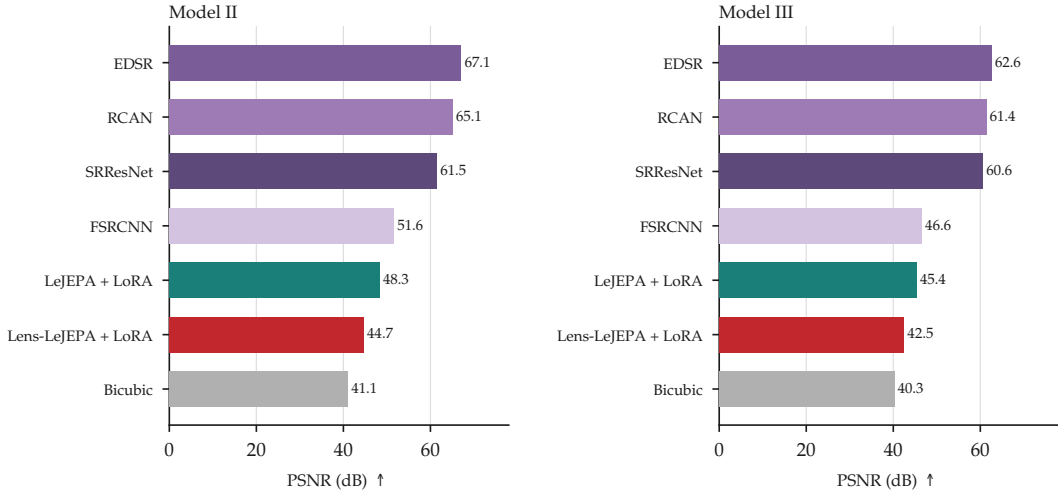


Figure 10: **Super-resolution PSNR on both test sets.** Frozen self-supervised features with a light decoder beat bicubic interpolation on both instruments; dedicated supervised networks trained from scratch under the same 50-epoch budget are stronger still.

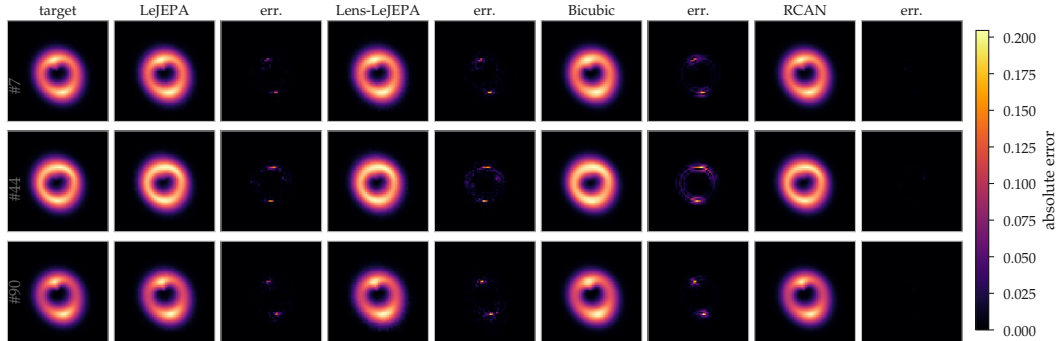


Figure 11: **Super-resolution examples on held-out Model III lenses,** in the same format as the Model II panel of Figure 3. The behaviour matches Model II, with slightly larger residuals reflecting the sharper HST-like arcs.

H Limitations

All three datasets are `lenstronomy` realisations with a known point spread function, known noise and a balanced, complete label set. Real survey images differ in all three respects, and additionally

carry selection effects, blending, and lens galaxies whose light must be subtracted. Nothing here tests that transfer. This bounds every claim in the paper, and it is the obvious next step: the natural experiment is to pretrain on unlabelled survey cutouts and evaluate on the small set of confirmed real lenses. Proposition 1 makes the patch permutation exact, but the ViT’s positional embeddings are not D_4 -symmetric, so the encoder is only approximately equivariant at convergence, with a small residual (Figure 5). A hard-equivariant backbone would remove that gap and is untested here.

I Reproducibility

Every number in this paper is read directly from the recorded held-out evaluation files by using the code provided.